(2)

ANNUAL REPORT

VOLUME 4

TASK 6: PARALLEL FUNCTION PROCESSOR DEVELOPMENT

REPORT NO. AR-0142-90-001

July 19, 1990

GUIDANCE, NAVIGATION AND CONTROL
DIGITAL EMULATION TECHNOLOGY LABORATORY

Contract No. DASG60-89-C-0142

Sponsored By

The United States Army Strategic Defense Command

COMPUTER ENGINEERING RESEARCH LABORATORY

Georgia Institute of Technology

Atlanta, Georgia 30332 - 0540

Contract Data Requirements List Item A005

Period Covered: FY 90

Type Report: Annual

DISCLAIMER

DISCLAIMER STATEMENT - The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

DISTRIBUTION CONTROL

(1) DISTRIBUTION STATEMENT - Approved for public release; distribution is unlimited.

(2) This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227 - 7013, October 1988.

1. Introduction

The DETL (Digital Emulation Technology Laboratory) simulation hardware centers on the development, implementation, and use of the Parallel Function Processor (PFP). The PFP is a 64 processor digital computer for use in computationally intensive applications that can be partitioned into functional blocks. The processors are grouped in two 32 processor clusters running from one common host. Each 32 processor cluster is connected by a crossbar switch. All inter-processor communication takes place over the crossbar(s). Simultaneous transfers may take place independently and switch patterns may be changed every cycle. In order to program the machine correctly, all inter-processor communication and data transfer lengths must be known beforehand.

The PFP has been designed to accomodate "hardware in the loop" simulations running in real time. Actual hardware components may first be simulated on one or more processors and later replaced with actual hardware interfaced to specified crossbar ports. The inputs and outputs to/from the device will appear identical to those it would see in an actual system.

Figure 1 illustrates the basic PFP architectural concept. Figure 2 illustrates a front view of the actual machine. A deeper level of architectural detail can be found in the final report for FY89, contract number DASG60-85-C-0041, volumes 1 and 2 [1].

1.1. Objectives

Within DETL, there are two main hardware systems: The PFP and the Seeker Emulator. (The Seeker Emulator is covered in volume 2 of this report.) The two systems are designed to function together as a simulation/emulation facility for kinetic energy weapons systems. The principal objectives of the DETL are as follows:

- Provide facilities for 6-DOF KEW emulation
- Provide real-time capability in excess of 2000 Hz
- Provide support for nonlinear functions
- Provide real-time emulation of IR FPA seekers
- To provide a facility for testing and verification of GN&C

Real-time emulation of IR FPA seekers is primarily the responsibility of the Seeker Emulator. The other objectives are primarily the responsibility of the PFP.

1.2. Requirements

The primary requirements for the period covered by this report fall into two categories. The first category centers on taking the existing system and readying it for possible delivery to remote sites. (Huntsville's KDEC facility and AEDC's LETS facility are the two mentioned the most often). This area includes user documentation, manufacturing documentation, training sessions, development and documentation of acceptance tests, reliability testing, and thermal testing. The second category centers on new architectural

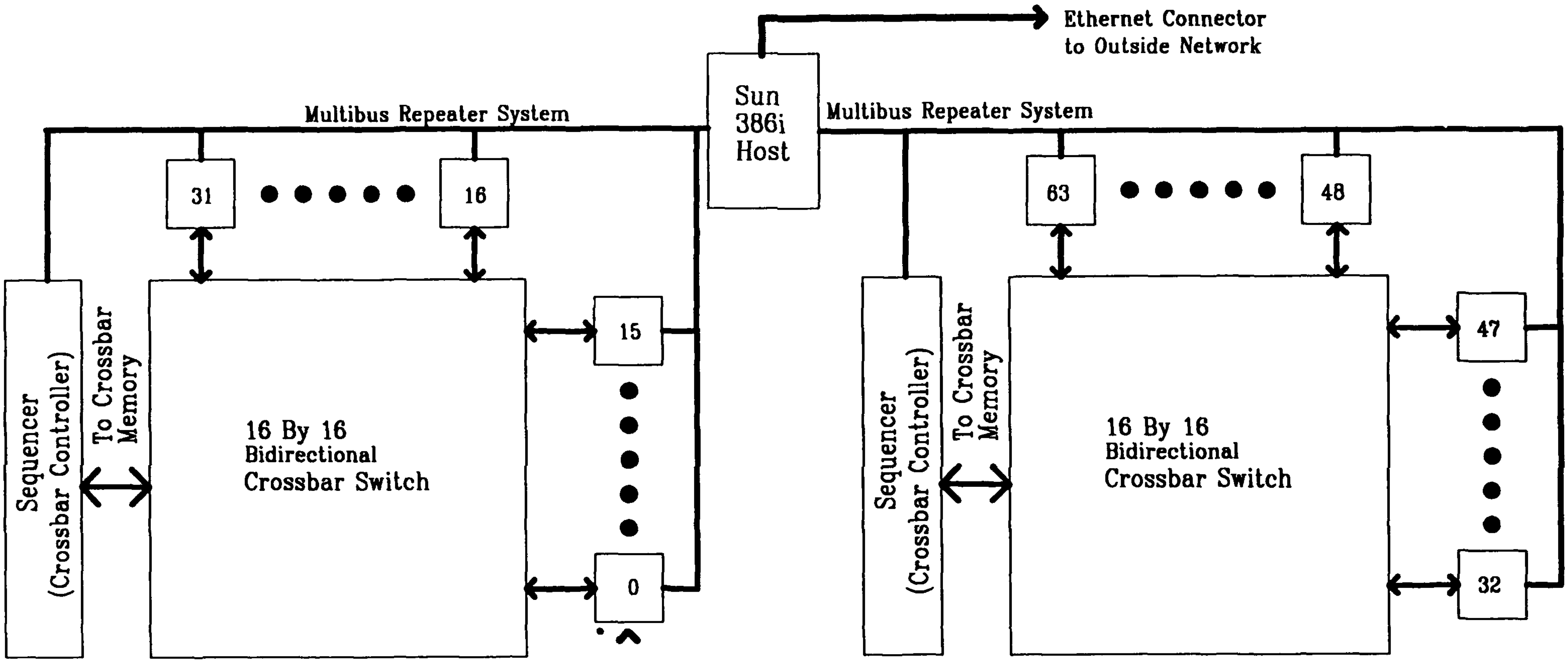


FIGURE 1. BASIC PFP ARCHITECTURE

developments for the PFP. This has included integration and support for new processors, bus structures, and new interfaces. Preliminary investigations have also been started for the development of a new more flexible interconnection network to replace the existing crossbar/sequencer combination.

The milestones met for the period covered by this report are as follows:

- Development and delivery of a complete set of documentation describing the requirements for manufacture and acceptance testing of the PFP so that the units can be reproduced.

- Development and delivery of a hardware operator's manual for the PFP.

- Development and delivery of a software programmer's manual for the PFP.

- Development and execution of a PFP training session.

- Development and execution of PFP reliability and thermal testing.

- Development and implementation of a materials management system for ordering PFP parts at the system, subsystem, and component levels.

- Expanded memory addressing capability on the Multibus I repeater system.

- Integration of the iSBC386/12 Multibus I based processor.

- Manufacture of a PFP unit for delivery to KDEC.

- Development of a Multibus II based PFP unit with support for iSBC386/120 and iSBC486/125 processors.

2. PFP

2.1. System Documentation

2.1.1. Technical Data Package

A four volume technical data package has been developed and delivered to USADC [2]. The package describes the requirements for manufacture and acceptance of the PFP. Volume 1 is titled "System Documentation". It contains all text on PFP assembly and each sub-assembly. Each sub-assembly is organized as a separate "User's Guide", including theory of operation, hardware options, assembly instructions, and programmable device listings.

Volume 2 is titled "Assembly Drawings". It contains all system level drawings and parts lists including all AC and DC chassis wiring, mechanical fabrication drawings, cable construction diagrams, subsystem placement, and all miscellaneous drawings.

Volume 3 is titled "Schematics". It contains all electrical schematics, assembly drawings, and parts lists for the circuit boards in the system. Each board is considered a sub-assembly.

Volume 4 is titled "Test Programs". It contains printouts of all PFP system and subssystem diagnostic

and acceptance tests.

2.1.2. PFP Hardware Operation Manual

A PFP Hardware Operation Manual has been written and delivered to USADC [3]. The purpose of the manual is to give the PFP operator a functional understanding of how the PFP works, its capabilities, and how to use it. In addition, the manual explains how to run system diagnostics and how locate errors based on the results. The PFP also contains two displays to aid in program debug and troubleshooting, 1) the crossbar status displays, and 2) the sequencer/processor transition boards. The manual goes through three examples showing how to read the displays and how to track programming bugs to a processing element based on what the displays read.

The first version of the manual has been delivered to USADC as a special technical report. The manual reflects the current system configuration. Since the PFP systems are continually being improved, minor changes to the manual are inevitable. For example, the current manual descibes an Intel 310 computer as the host. The Sun 386i host is fully functional from a hardware standpoint, but the system software support for it is not yet complete, thus Intel 310 is still the main host in use. A final version of the manual is due in 1991 (See Section III Schedule/Milestones for details) which will reflect the current configuration as of that date, as well as changes deemed necessary from feedback on the original manual.

2.1.3. PFP Programmer's Manual

A Programmer's Manual for the PFP has been written and delivered to USADC [4]. The manual provides the information needed for a programmer to understand and program the Parallel Function Processor. Information on languages, syntax, and memory limits are presented. Additional information on how to use existing system software is also discussed.

The first version of this manual has been delivered as a special technical report. As with the hardware operation manual, a final version is due in 1991. The current version assumes the Intel 310 as host and the Intel family of processors as the processing elements. The final version should use the Sun 386i as host and contain information needed to use the GT-FPP/3 as a processing element, with both the ADA and C programming languages supported. Any changes deemed necessary from feedback on the original manual will also be included.

2.1.4. Materials Management System

Purpose

The purpose of the materials management system is to provide an organized, automated way to order PFP parts at the component, subsystem, and system levels. To do this, a database has been built using Borland International's Reflex product. The database is used to accumulate all needed parts for a particular PFP setup. All parts from each sub-assembly are summed into one ordering list for the purpose of ordering all similar parts together. This way less parts orders are generated the possibility of overlooked or duplicated parts is reduced.

Database form and Contents

The database is made up of sections called FIELDS. The FIELDS are shown in Table 1.

Table 1 Fields Used in Materials Management System

SUBASSEMBLY	Board or hardware piece name.
SUBASSE QTY	Quantity of sub-assembly per setup.
QTY PER ASSE	Quantity of part per sub-assembly.
PART NUM	Part number of particular part.
REFERENCE NUM	Reference number used in Technical Data Package.
VENDOR	Vendor who sells the particular part.
MANUFACTURER	Manufacturer of the particular part.
SS	Specifies if part is sole sourced or available from multiple vendors.
ITEM DESCRIPTION	Description of particular part.
ENGINEER	Engineer responsible for board or hardware piece.
UNIT PRICE	Price per unit piece.
TOTAL COST	Cost for multiple parts per 1 sub-assembly.
EXTENDED COST	Cost per multiple number of sub-assemblies.

Organization

The fields have been organized into two output formats, the PFP Materials List and the PFP Purchasing List. The formats are designed around specific output requirements.

The PFP Materials List format is used in the documentation process. It is formatted to best show item or part information as referenced in the Technical Data Package. The Technical Data Package contains a parts list in this format for each assembly and sub-assembly in the PFP.

The Purchasing format is used in the purchasing process. It is formatted to sort all similar parts together, sort the vendors, and add the grand totals for a projected system cost .

Database Use

The Reflex Database system is a straightforward, easy to use, flat database. All the needed subassembly and parts breakdown are already intact and may be manipulated as shown in the following examples.

A. In order to get a materials list for documentation and board manufacturing purposes do the following:

1. Choose the specified subassembly parts by using the filter command.

2. Fill in the SUBASSE QTY (subassembly quantity) with the appropriate number.

3. Print the contents generated by steps 1 and 2 in the PFP Materials List Format.

B. In order to accumulate all needed parts and prices for the purchase order process use the database contents generated in A1 and A2 and print contents in the Purchasing Format.

Database Modification

The Modification of parts, prices, and quantities in the database requires the following steps.

1. Filter by PART NUMBER to find the part/s needing to be changed.

2. Make modifications then remove filter.

3. Repeat steps 1 and 2 until all modifications are made.

4. Save the modified database.

The process required to add new parts to the database is as follows.

1 Press the end key. This will bring cursor to a blank at the end of the database.

2. Enter new data into all displayed fields.

3. Repeat steps 1 and 2 until all new data is added.

4. Run the Sort command.

5. Save the modified database.

The process required to delete unwanted parts from the database is as follows.

1. Filter by PART NUMBER in the List View to find the part/s needing to be deleted.

2. Press F3 to select row containing about to be deleted information.

3. Press the delete key.

4. Repeat steps 1-3 until all specified parts are deleted.

5. Save the modified database.

2.2. PFP Training

A 3 day PFP training course was held in December. The course was attended by 7 people. The course was divided into four separate sessions. Each session was followed by a short examination.

Session 1 contained a brief introduction to the parallel function processing approach. Basic parallel programming techniques were presented, including the methodology needed for partitioning a problem into its functional blocks. A small problem was presented and partitioned as an example.

Session 2 covered the hardware operation of the PFP. This included a fuctional description of the basic PFP architecture and how it works, its capabilities, and how to take the functional blocks and map them onto the machine. The basic issues of how to turn both the PFP and host on and off, how to start it, and where to access mass storage were also covered. The session also explained how to interpret the displays that are part of the machine in conjunction with program debug and system troubleshooting.

Session 3 covered programming the PFP. Topics included development and compilation of code for processing elements - including the use of special purpose I/O routines to interface with the crossbar and host computer, development of crossbar code - including syntax and compilation, how to intergrate and load processing element codes with the crossbar code into a working program, and how to run the program.

Session 4 was held in the laboratory and consisted of dividing the attendees in groups of two and having them program two small problems on the PFP. The first problem was given in its complete form. The task was to copy it into the machine, compile it and run it. The second problem was given in block diagram form with the functional blocks outlined. Programmers developed their own code and ran it on the PFP.

The course size is limited by session 4. It requires that each group have enough access to the PFP to solve the programming problems in a reasonable amount of time. When using 1 PFP for training, the course size should be limited to around 10. No definite training schedule is planned. The course is available to be repeated when necessary.

2.3. PFP Testing

2.3.1. Reliability Testing and Temperature Analysis

A special technical report, "Parallel Function Processor Reliability Test" has been written and delivered to USADC [5]. The test consisted of running the PFP system diagnostics in an infinite loop, collecting thermocouple data from 32 different points on the system, and logging any system errors that occurred in an output file. Temperature data was collected over a 103 hour period. Plots of 31 of these points are included in the report, one thermocouple did not work correctly, giving all negative temperatures.

The system was run with 2 full crossbars, 2 sequencers, 1 array interconnect link (i.e., one board on each crossbar) and 45 processors. All processors were the GT-FPP/3 floating point processor, which uses the most power and generates the most heat of all the processors currently supported. The remaining 17 processor slots were empty. The right processor bank was fully populated (31 processors, one array interconnect) with the remaining boards (14 processors, 1 array interconnect) on the left side.

Four diagnostic tests, T3, FPPMU, T2, and FUNCTION were run in a continuous loop for the whole

week. Results are given in two forms, per test and per processor.

2.3.2. GT-FPP/3 Accuracy Analysis

The GT-FPP/3 Floating Point Processor contains 10 hardware assisted functions. The hardware assisted functions are supported through the use of an add on board called the GT-FFS/1. The hardware assisted calculations are carried out much faster than the software algorithms used on most machines, thus adding to the GT-FPP/3's high performance. The functions currently supported by the GT-FFS/1 are:

1. Sine
2. Cosine
3. Tangent
4. Arcsine
5. Arccosine
6. Arctangent
7. Exponential
8. Natural logarithm
9. Reciprocal
10. Square root

Two programs were written and executed on the Parallel Function Processor (PFP) for each function. Both programs compared the values calculated by the GT-FPP/3 - GT-FFS/1 combination to the values calculated by the Intel 310 host computer. The first test generated the absolute difference between the numbers. The second test generated a relative error, using the number computed by the Intel 310 as the correct answer.

Two graphs were made for each function, one for absolute error and one for relative error. Although no major discrepancies were uncovered, full interpretation of the results is not yet complete. The graphs are included in Figures 3 through 23. After fully finishing all interpretations, the full analysis will be submitted to USADC as a special technical report.

2.4. System Buildup

2.4.1. Integration of iSBC386/12 Processor

The Intel iSBC386/12 processor has been integrated into the PFP environment and is now available for use. The board is based on a 20 Mhz 80386 processor with an accompanying 80387 math coprocessor.

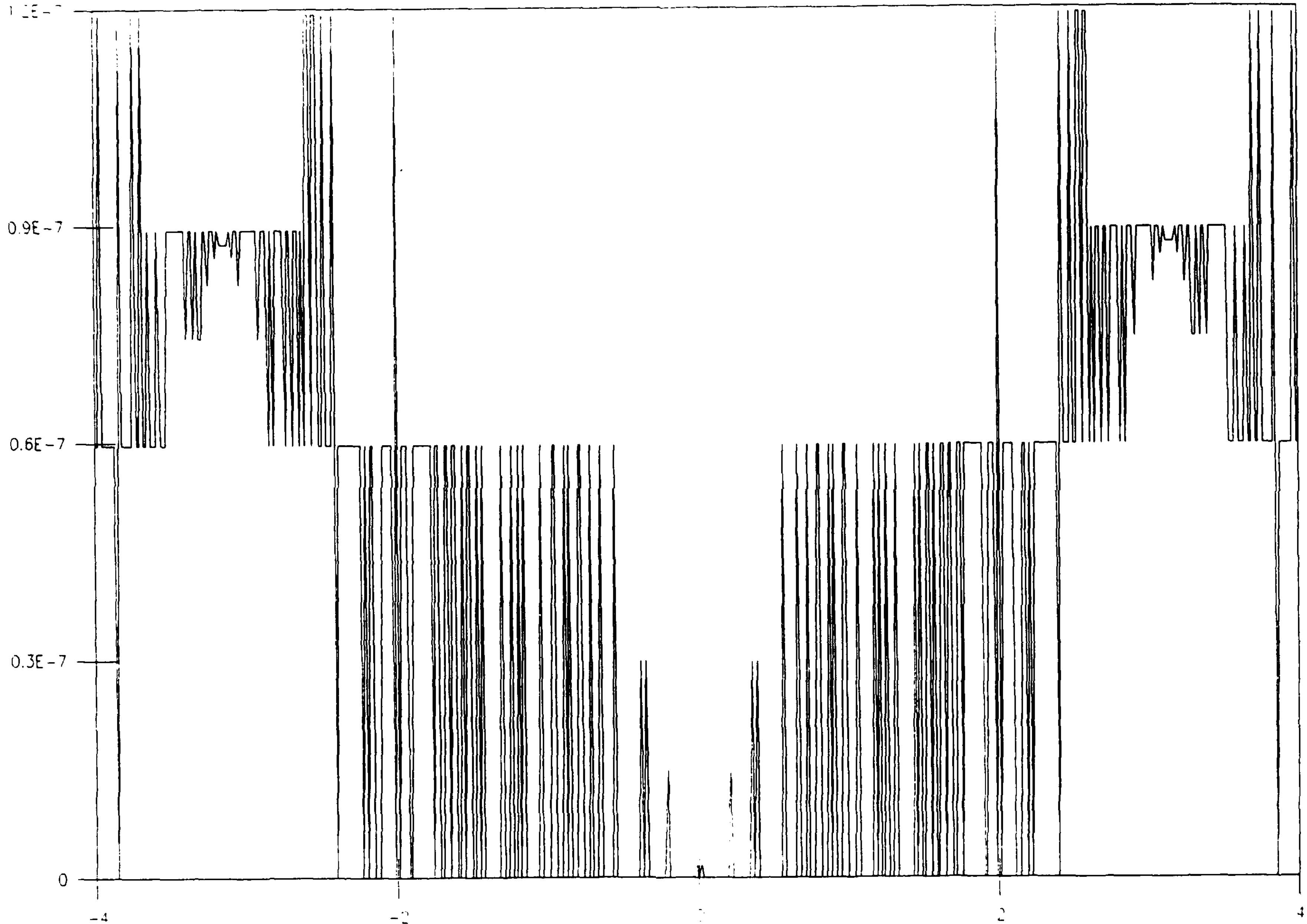


Figure 3. Absolute error in sine function over reference angles.

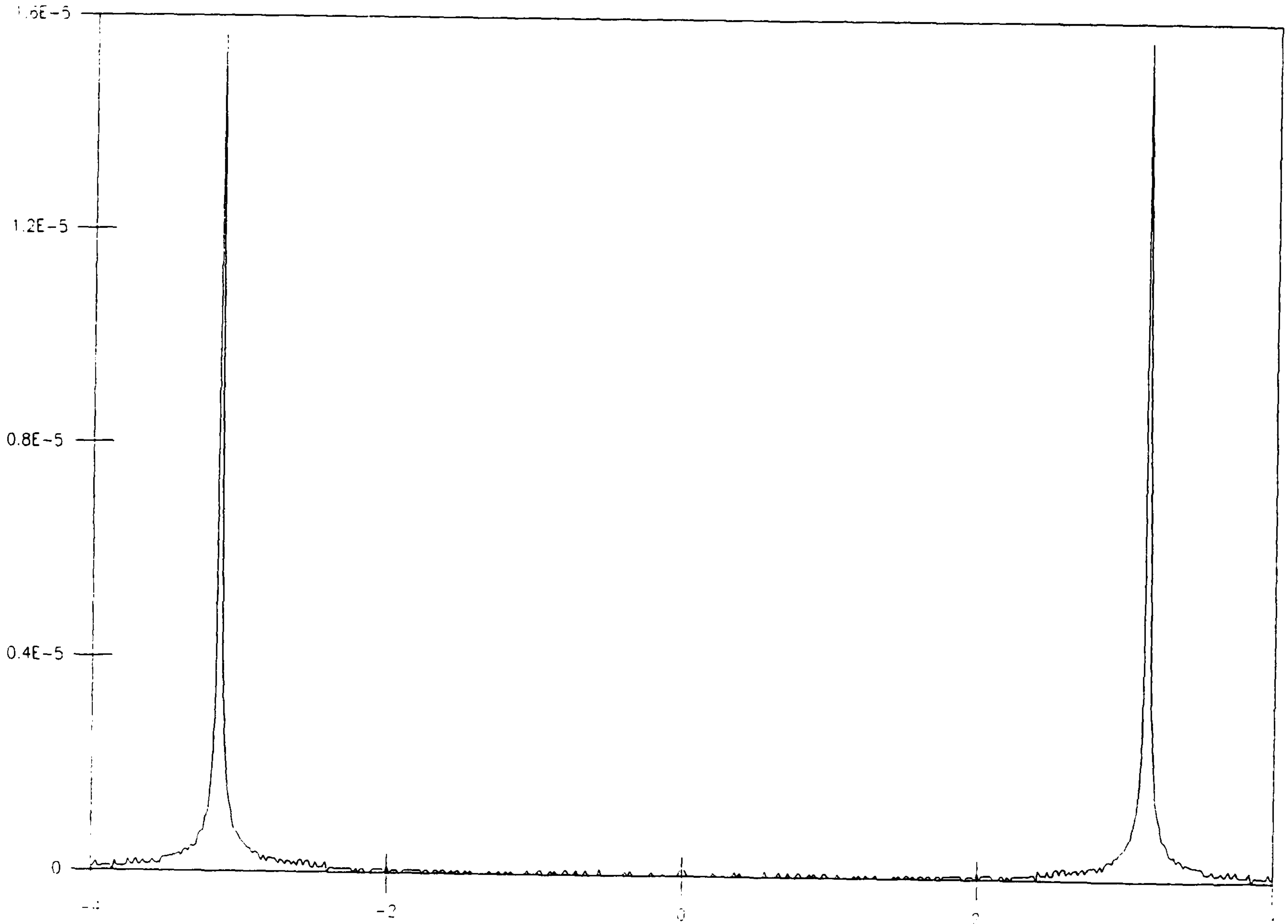


Figure 4: Relative error in sine function over reference angles.

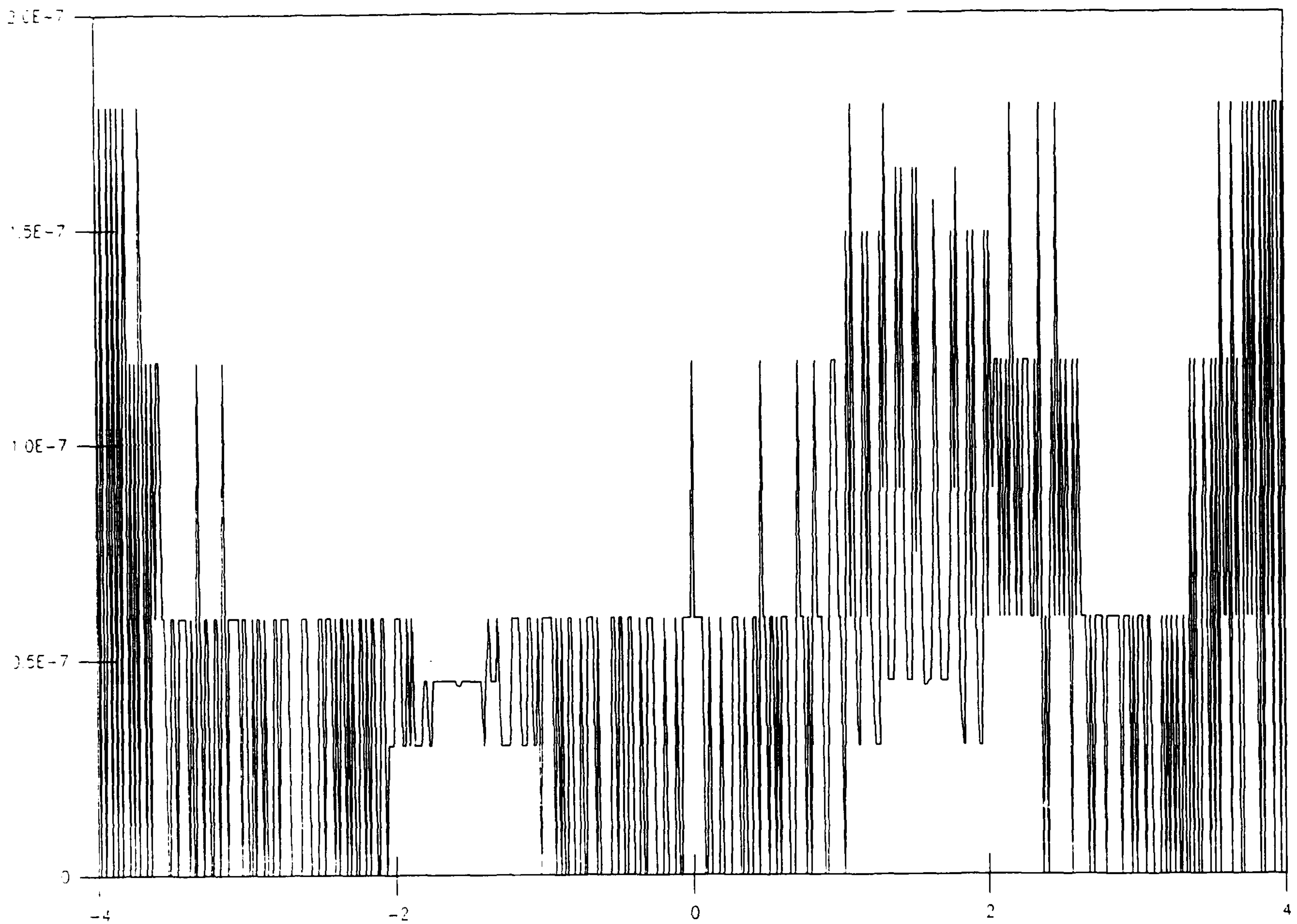


Figure 5. Absolute error in cosine function over reference angles.

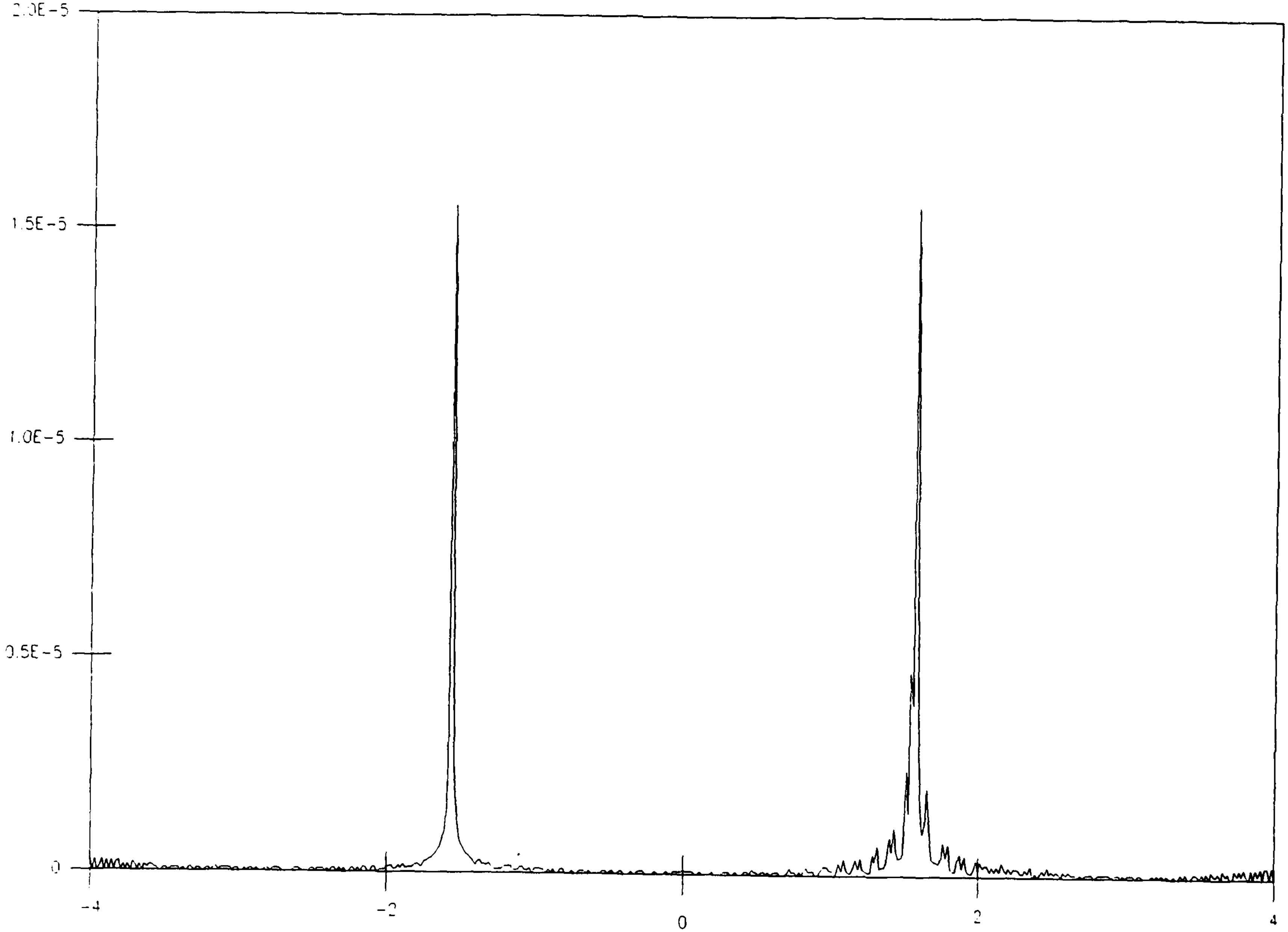


Figure 6. Relative error in cosine function over reference angles.

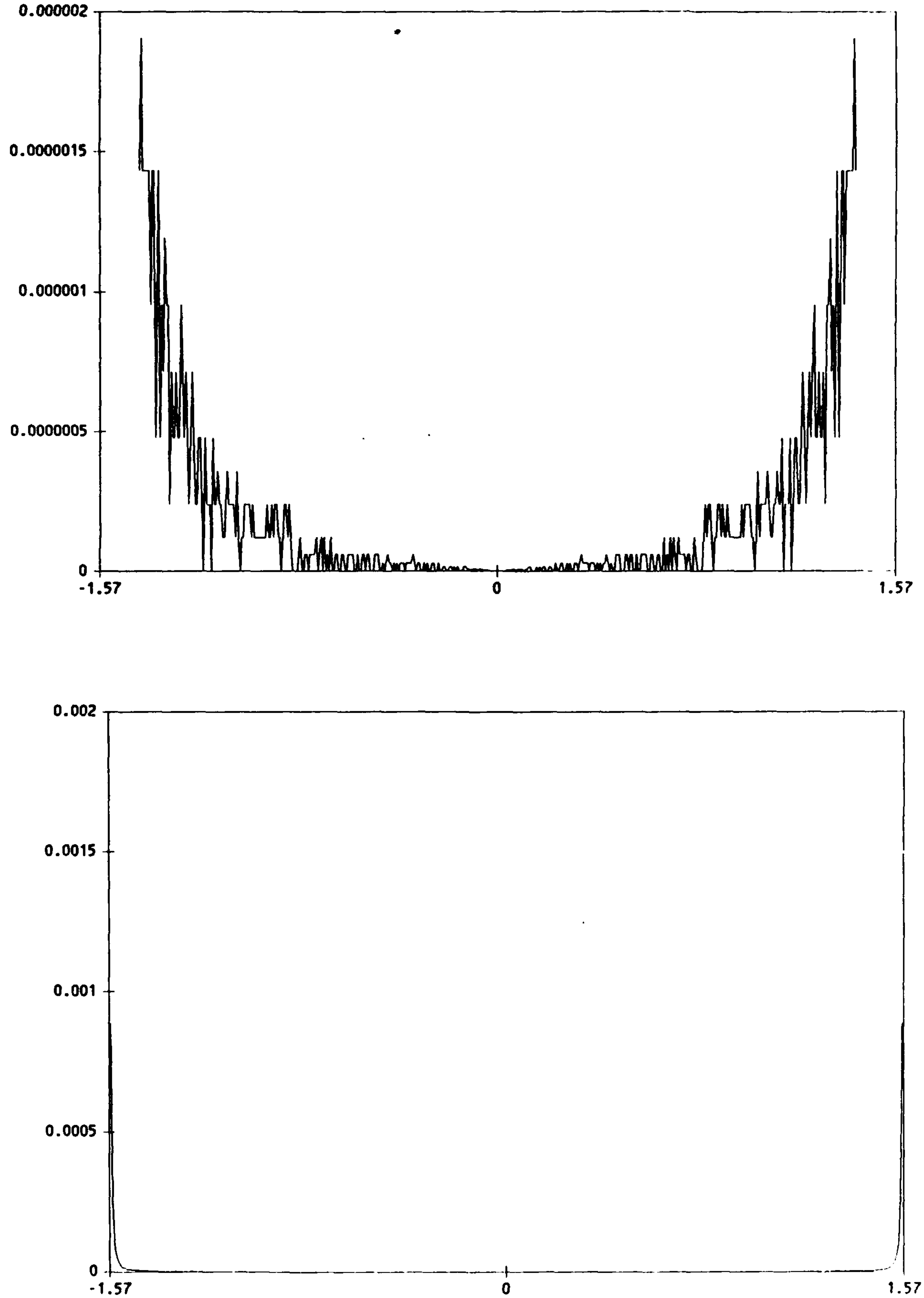


Figure 7. Absolute error of tangent function at reference angles (bottom) full view; (top) close up view of error.

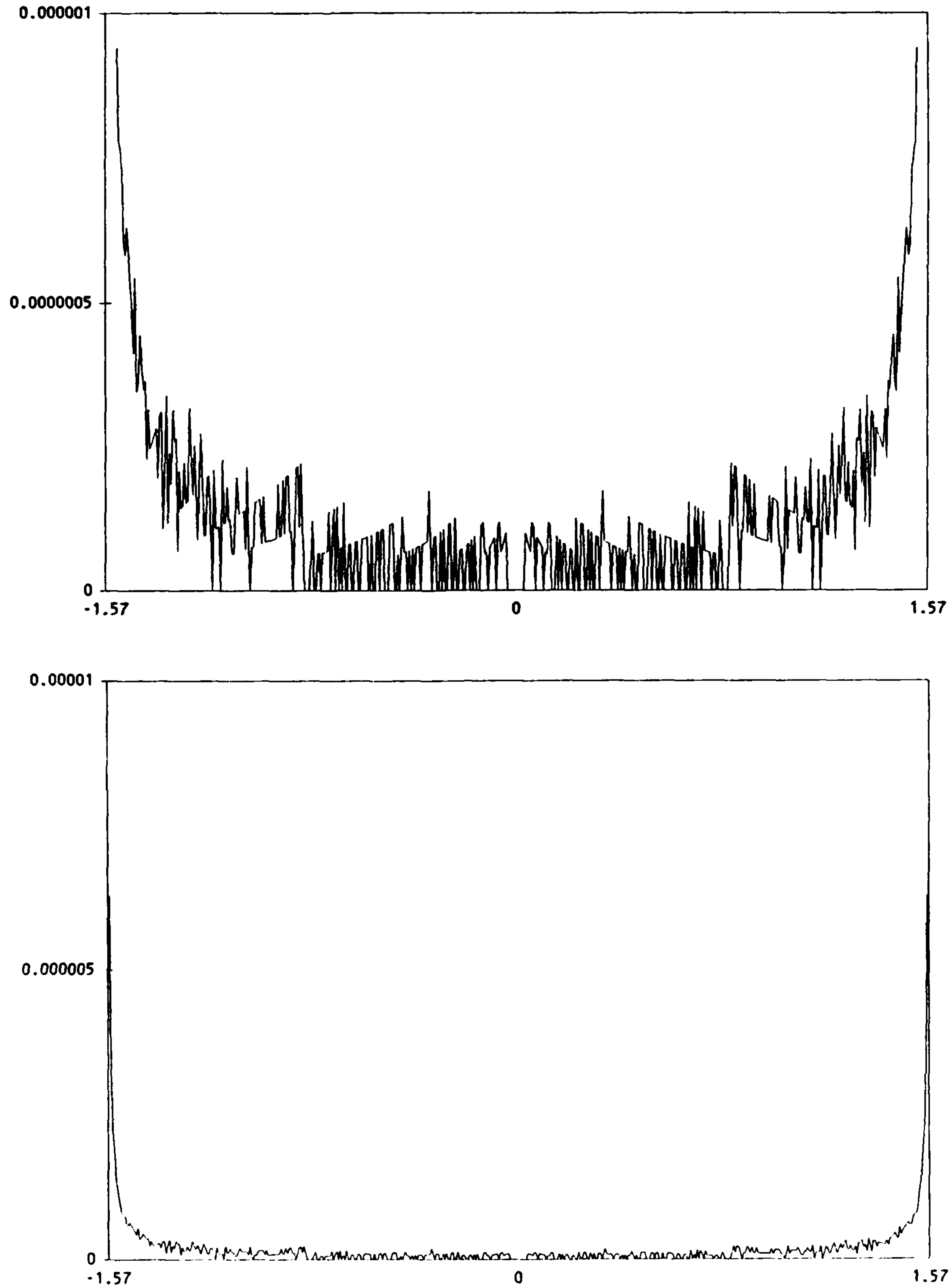


Figure 8 Relative error of tangent function at reference angles (bottom) full view; (top) close up view of error.

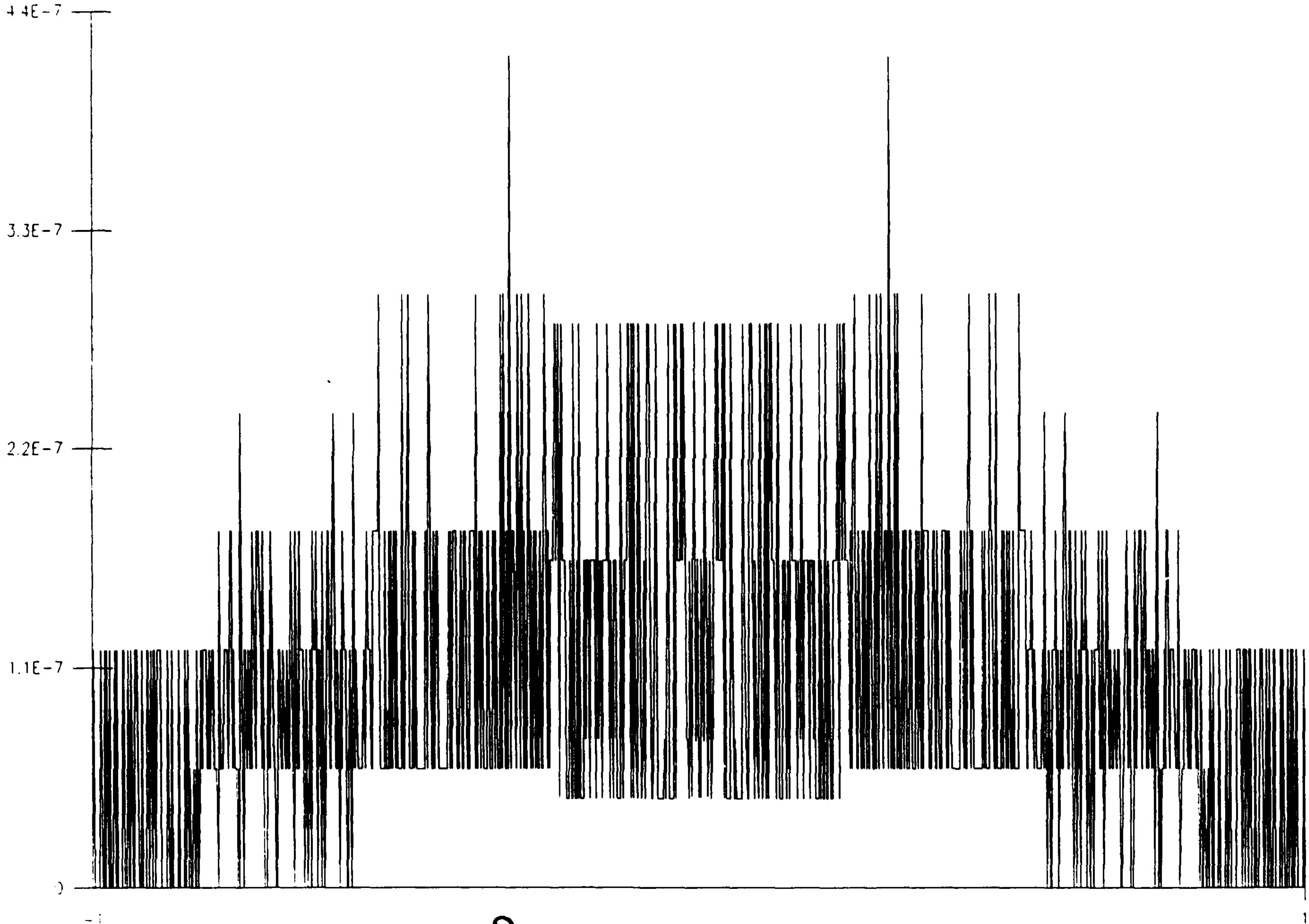


Figure 9. Absolute error in arcsine function.

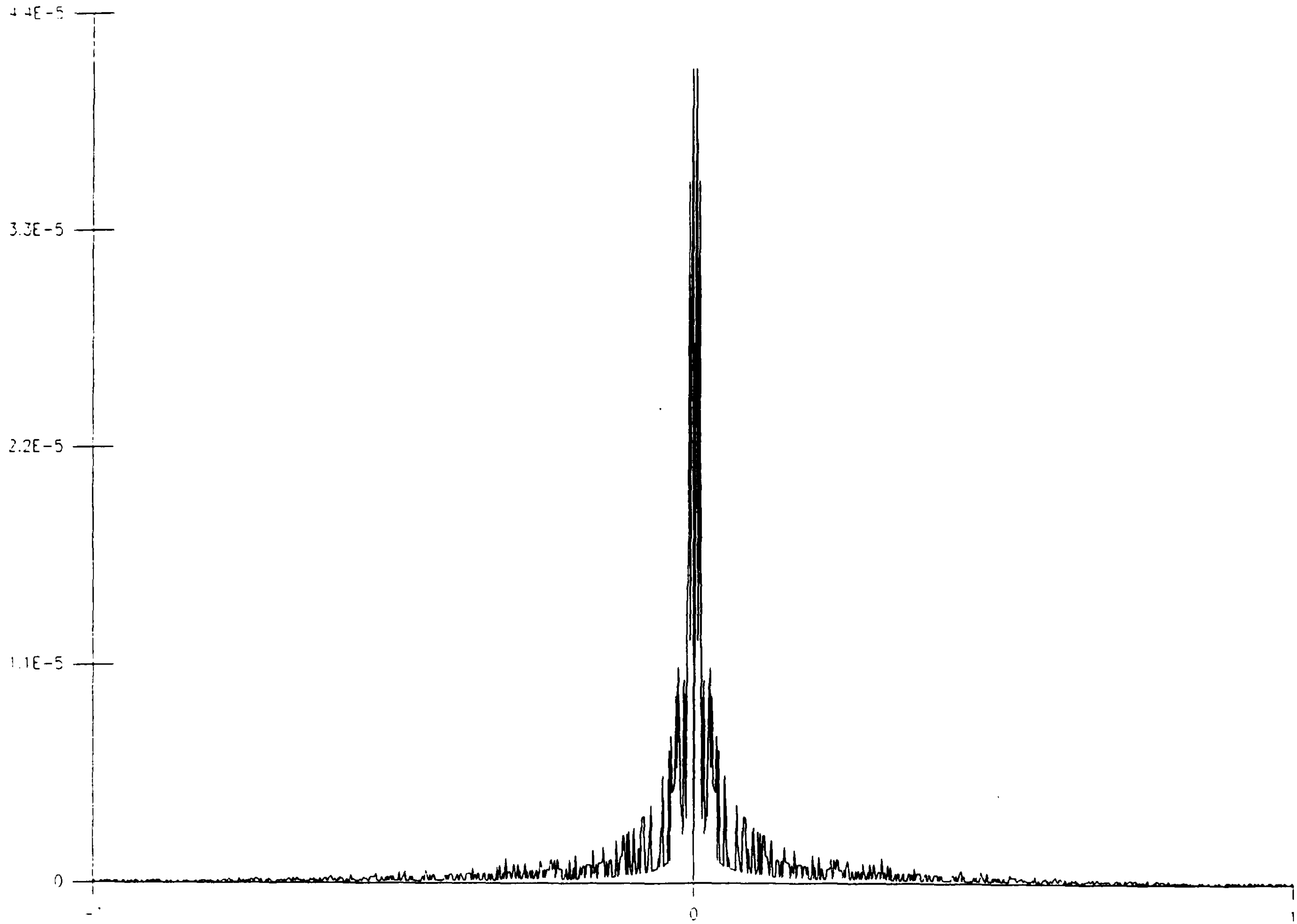


Figure 10. Relative error in arcsine function.

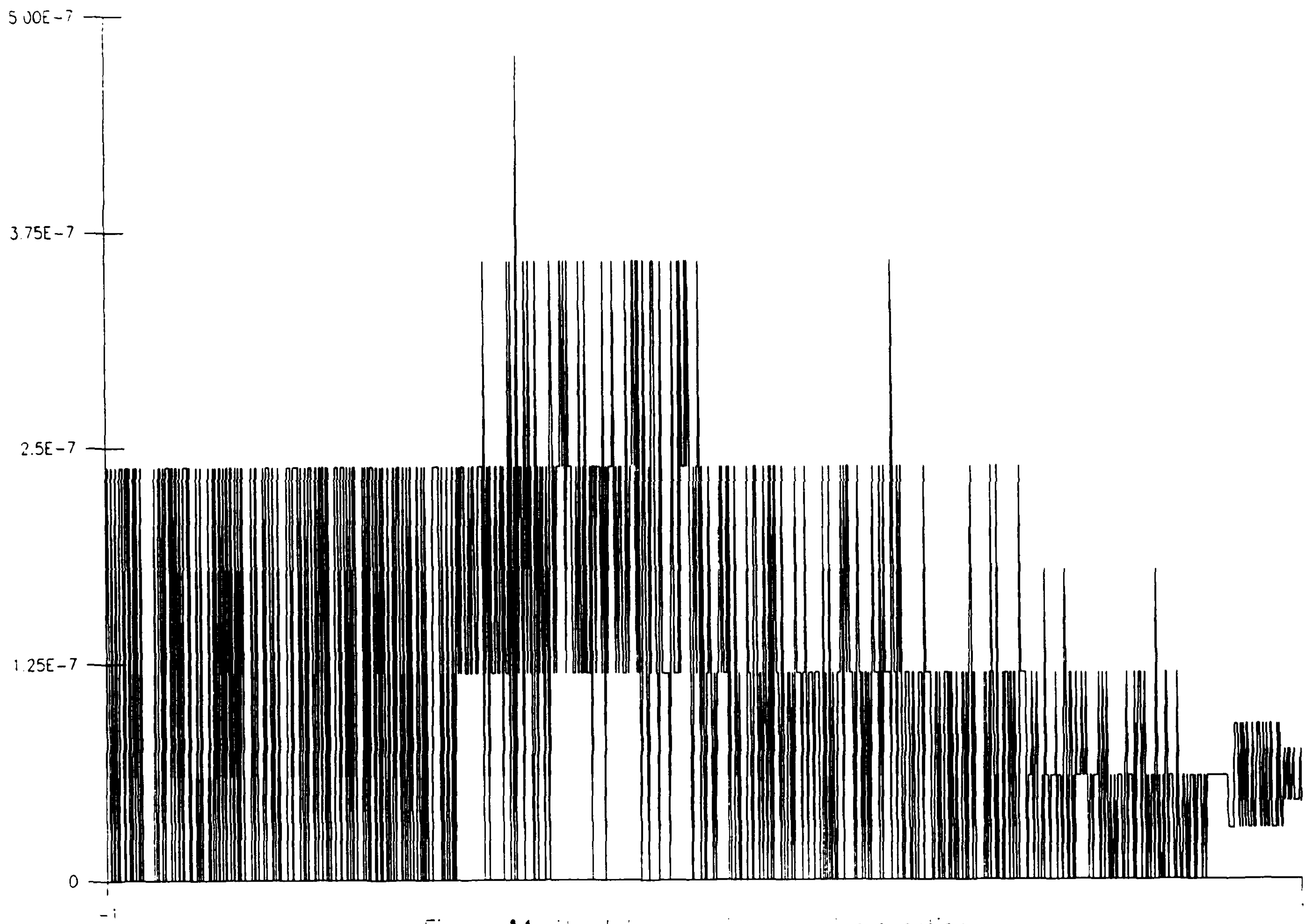


Figure 11. Absolute error in [illegible] cosine function.

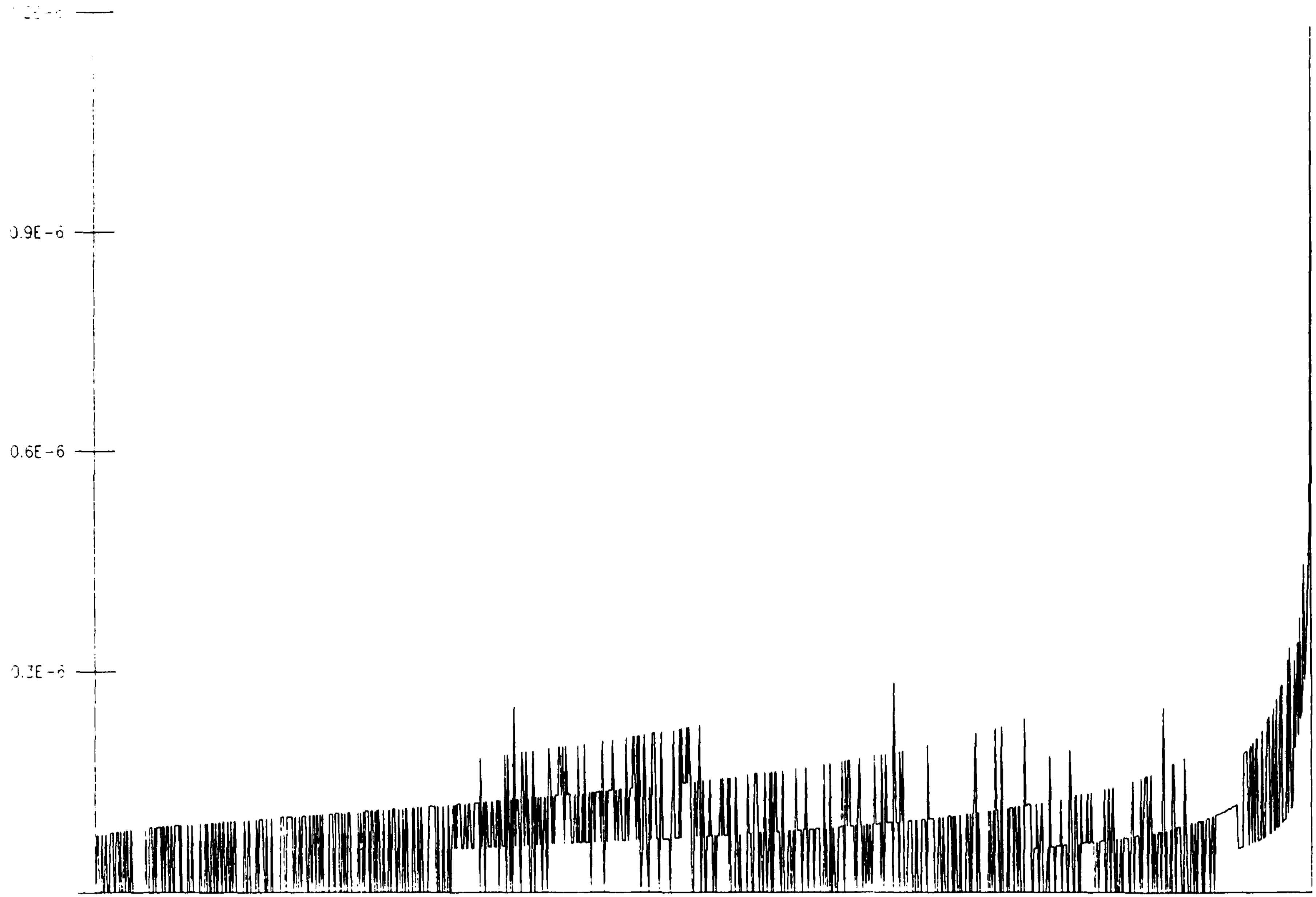


Figure 12 Relative error in arccosine function.

Figure 13 Absolute error in arctangent function.

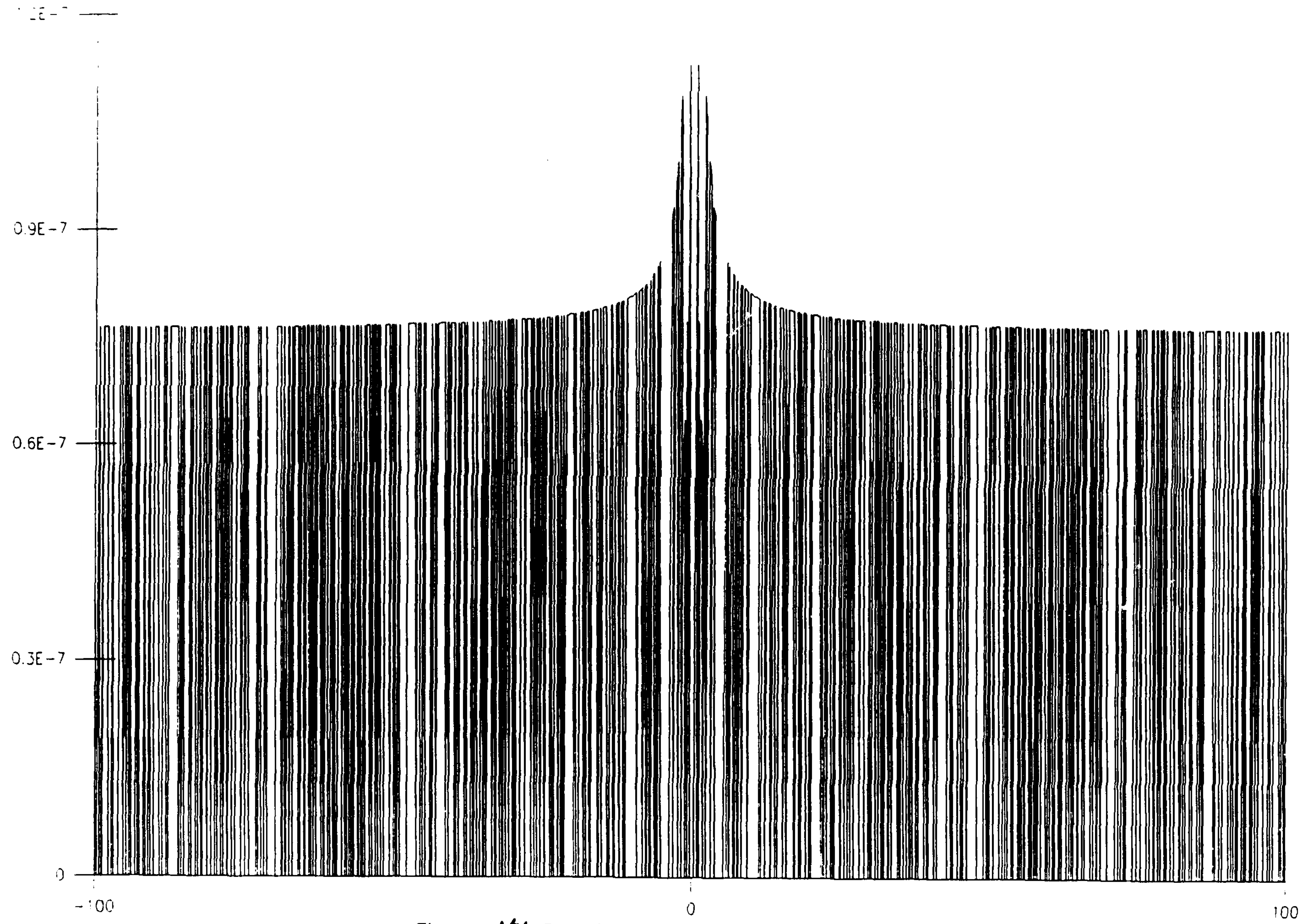


Figure 14. Relative error in arctangent function.

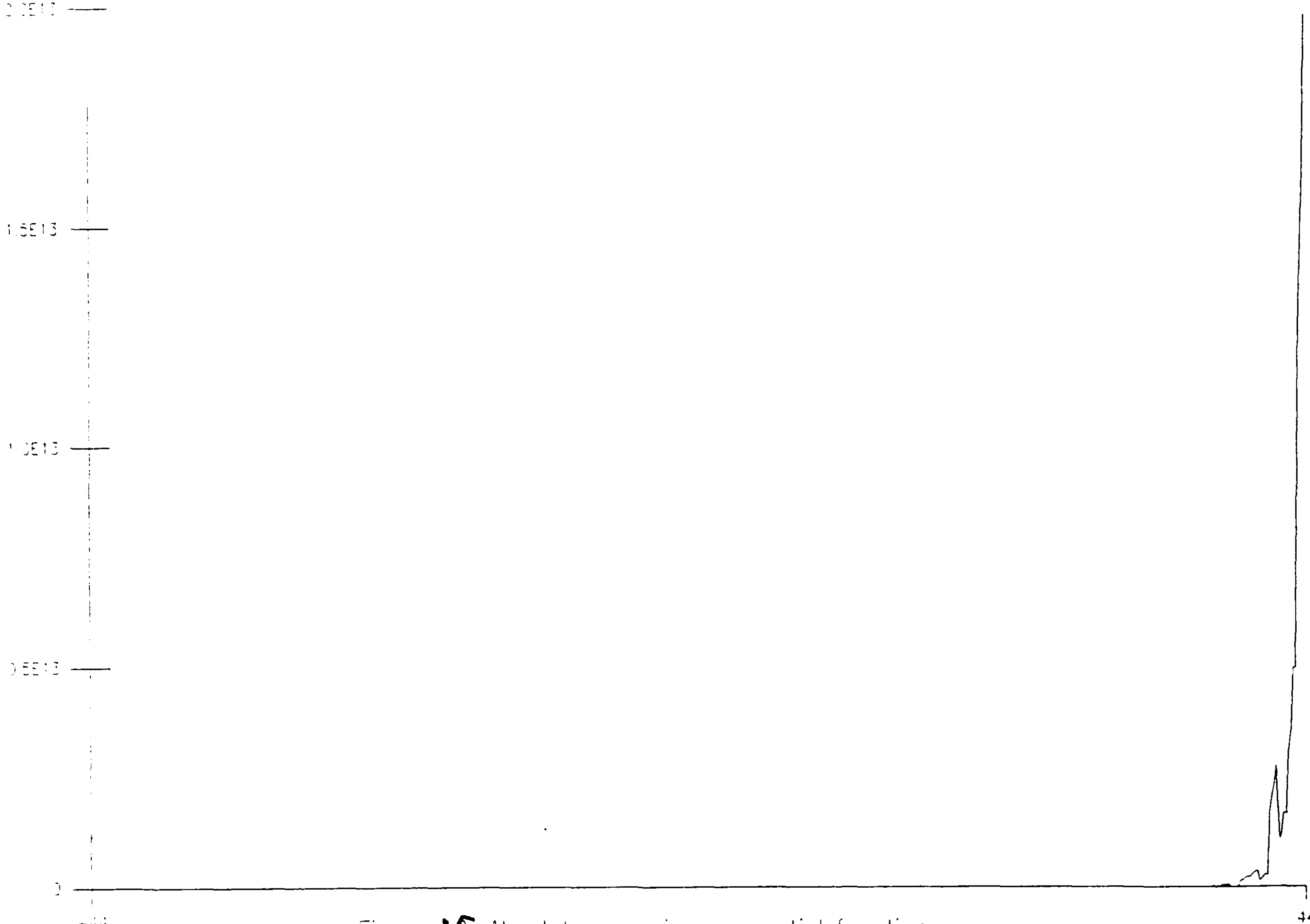


Figure 15 Absolute error in exponential function.

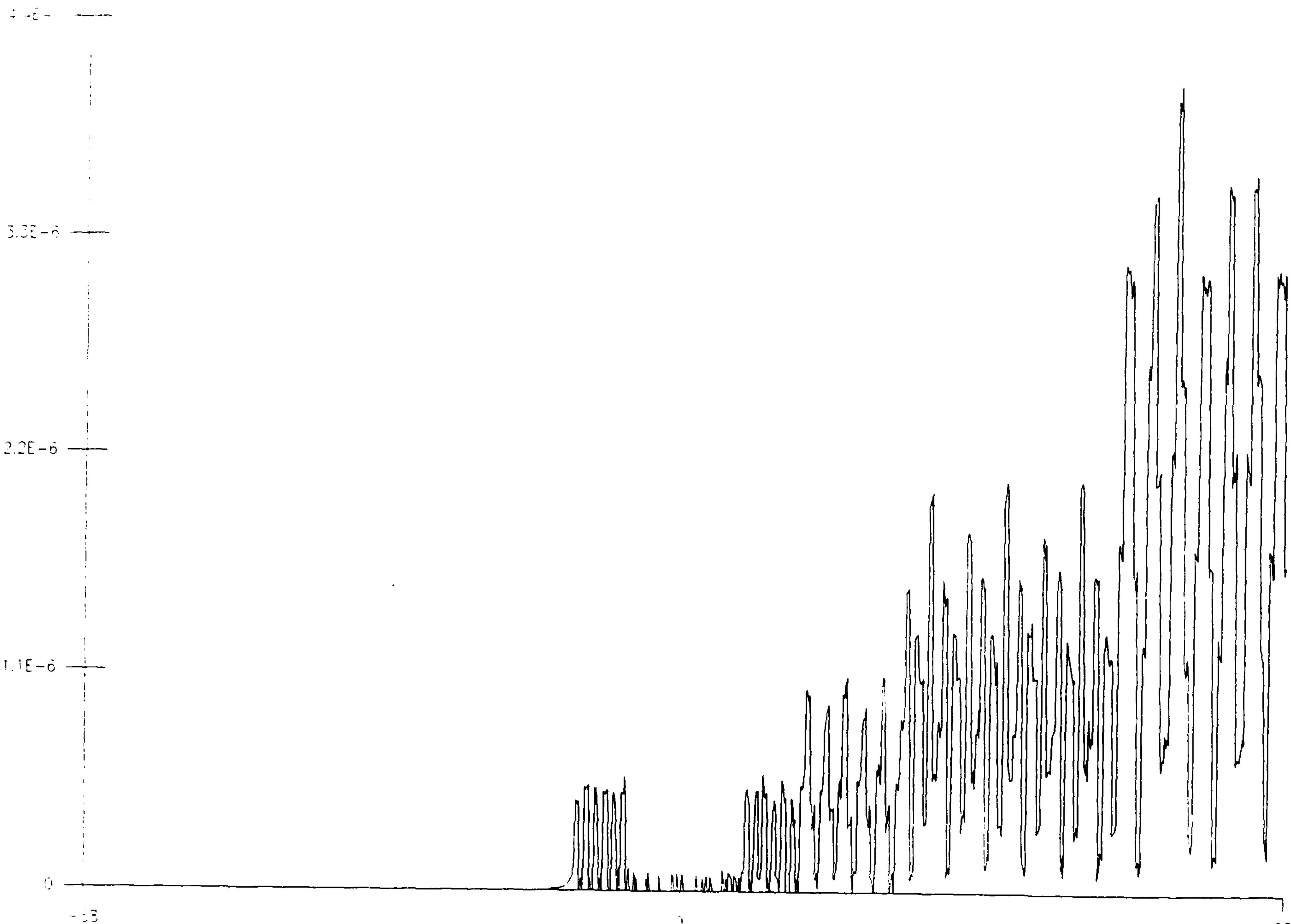


Figure 16 Relative error in exponential function.

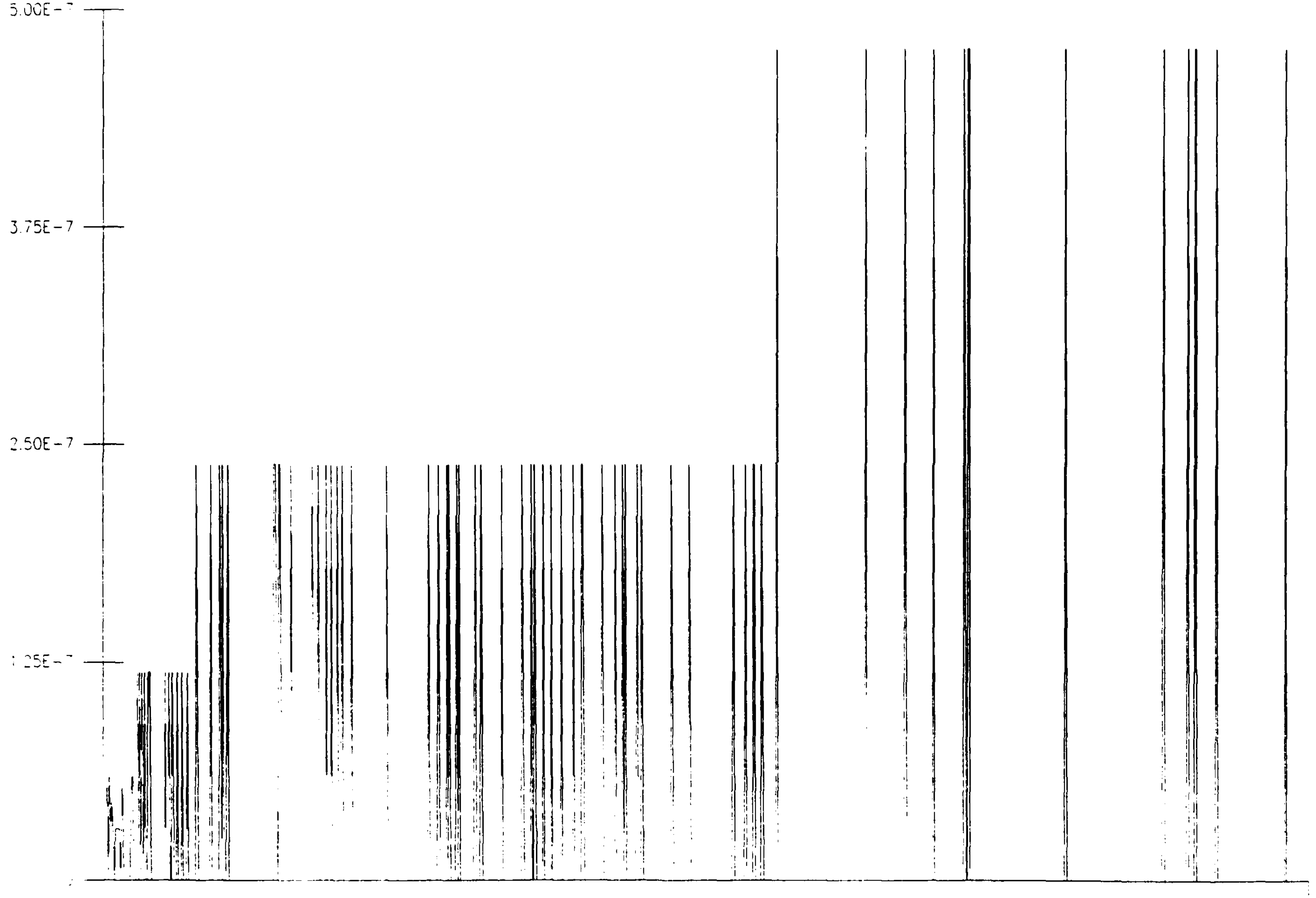


Figure 17. Absolute error in natural log function.

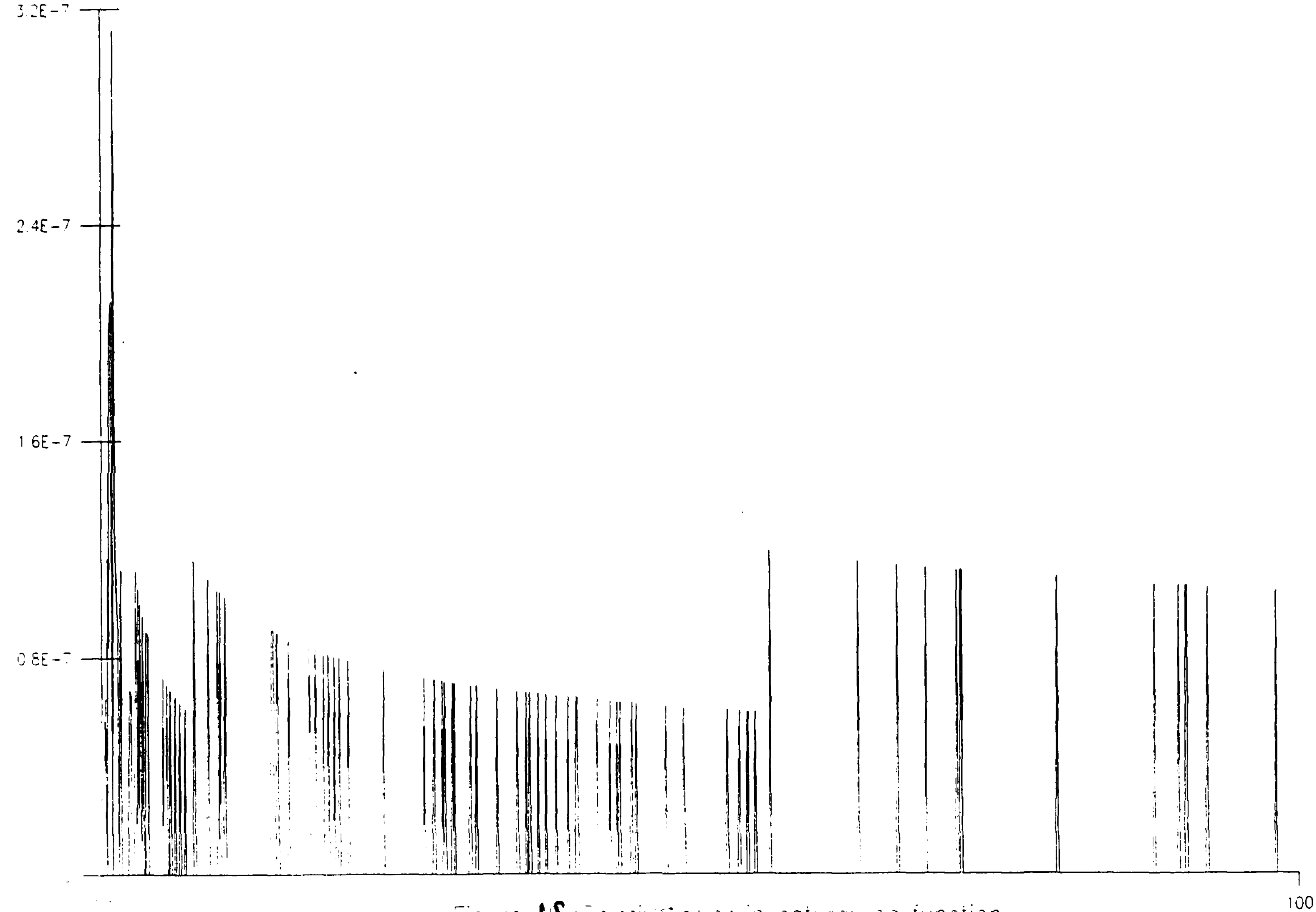


Figure 18 ~~Relative~~ Absolute error in natural log function.
OK

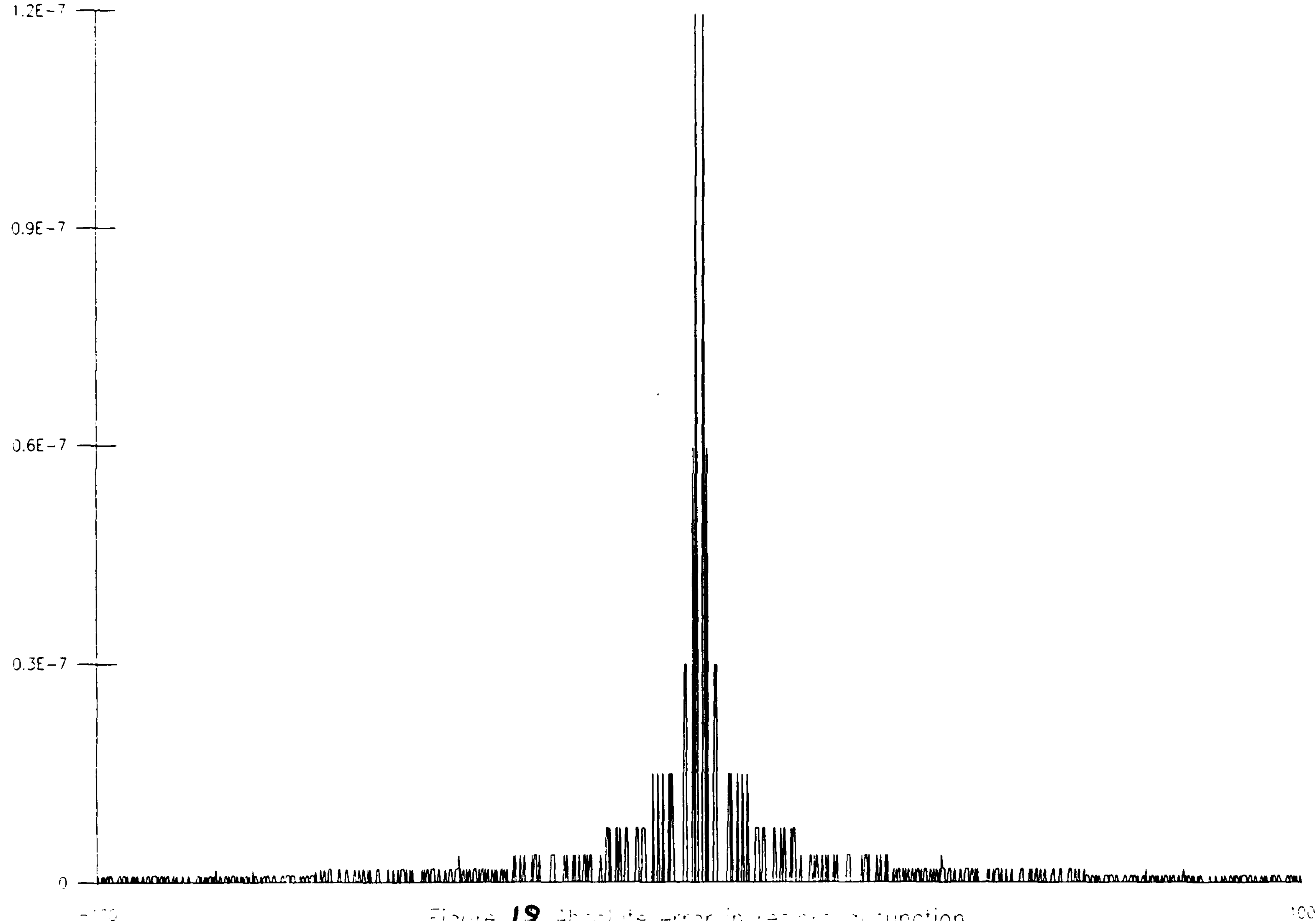


Figure 19 Absolute error in [illegible] function.

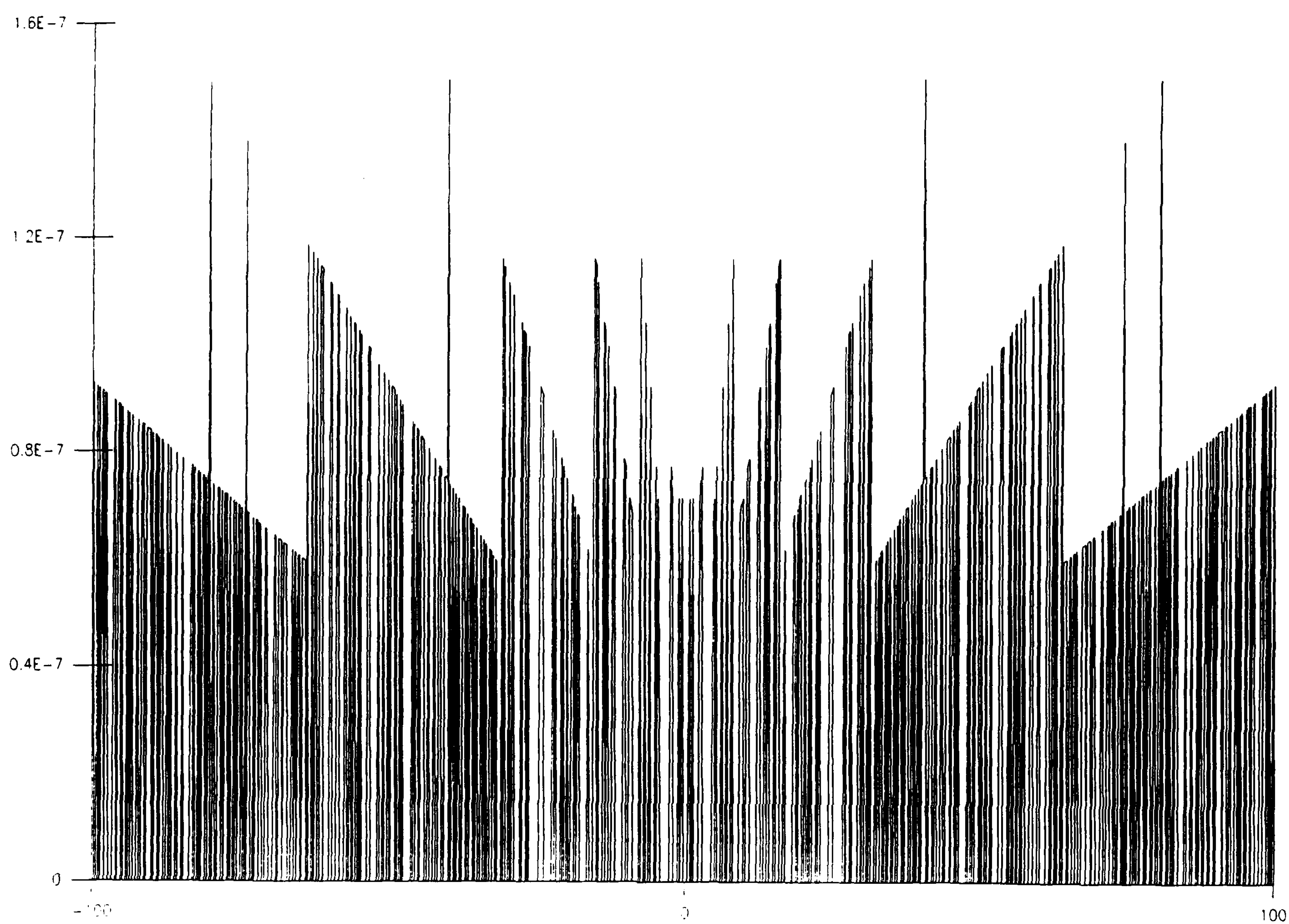


Figure 20 Relative error [illegible] function.

82

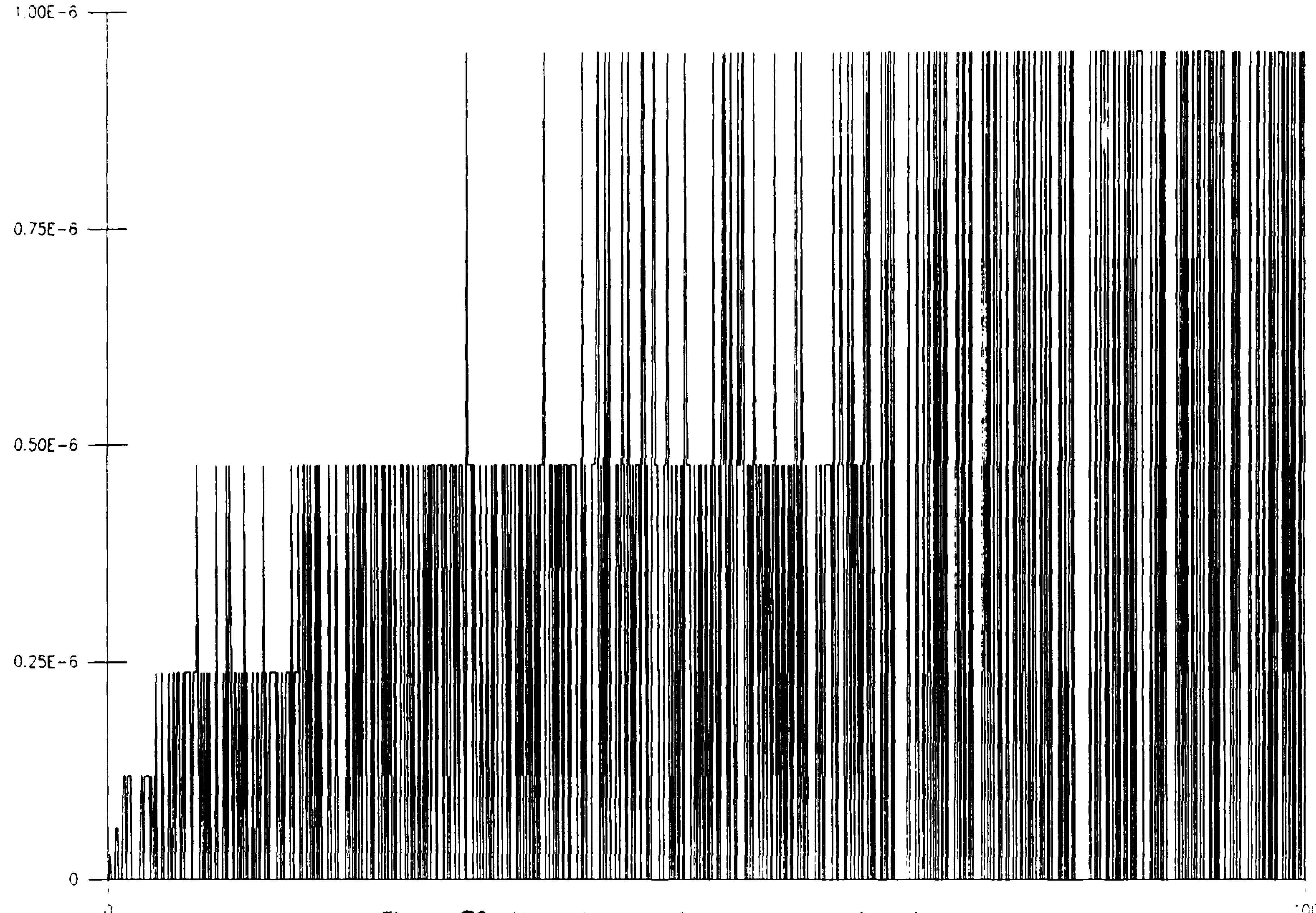


Figure 21. Absolute error in square root function.

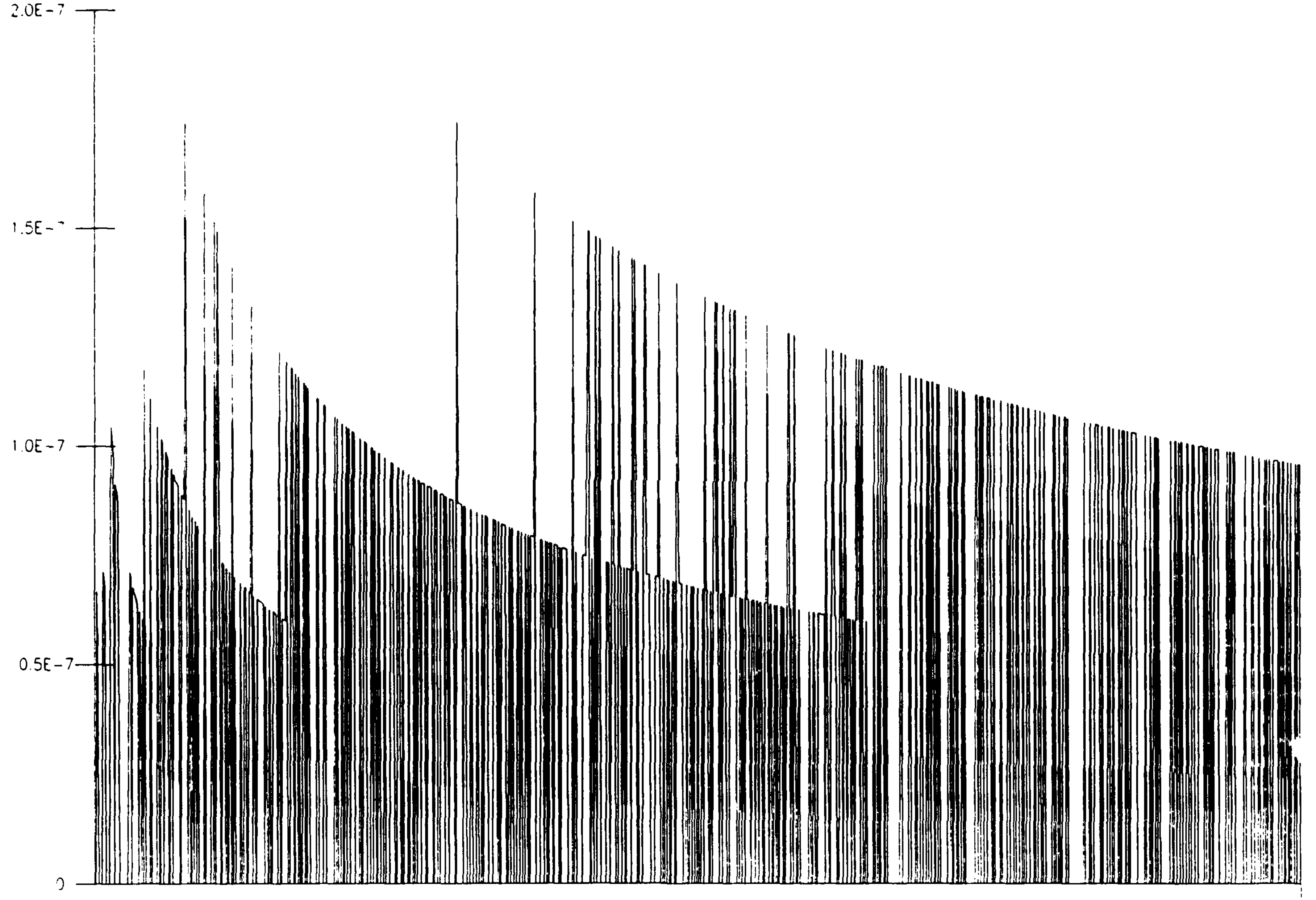


Figure 22. Relative error in square root function.

2.4.2. DETL PFPs

The CERL's Digital Emulation Technology Laboratory has two PFPs currently in use and one under construction. A software development PFP, hosted by a Sun 386i computer, is primarily being used for developing a C compiler for the GT-FPP/3 floating point processor. The unit currently contains 32 GT-FPP/3s. An Ada to C translator is being bought so that both Ada and C languages will be available for the board. Several Intel iSBC386/20 processors are available to interchange with the GT-FPP/3 boards. Software development is under way to support the 386/12 within the Sun host environment.

A PFP "test station" that can support up to 64 processing elements, originally put together from spare parts, is primarily used for testing new board assemblies, debugging defective assemblies, and for testing and integrating new PFP components. Currently, the unit is populated with 8 iSBC386/12 processors and is being used for EXOSIM development work. The system is hosted by an Intel 310.

The thirty-two processor system that has been located in the DETL's secure laboratory is currently being upgraded to match the configuration of the KDEC PFP. Processor card cages are being retrofitted to provde more address capability. The system will be fully populated with iSBC386/12 processors and eventually hosted with a Sun 386i computer. When completed, the EXOSIM development work will be moved back to this machine.

2.4.3. KDEC PFP

A PFP prototype has been built specifically for use by the KDEC facility. The unit is capable of supporting 64 processors and 2 full crossbars. The unit will initially be shipped with 32 processors and 1 crossbar. The other crossbar and 32 processors may be added on site at a later date. The system is hosted by an Intel 310 computer. Programming languages available include FORTRAN, Pascal, C, and PL/M.

The processors currently installed in the system are the Intel iSBC286/12 boards which have been used in the CERL laboratory for the past 2 years. These processors, and accompanying software, are fully debugged and will provide a stable environment for the KDEC programmers to learn how to use the system. The other 32 processors, to be installed later, could be either 80386 or 80486 based (both code compatible with the 80286 based processors) or the GT-FPP/3 floating point processors.

The system is complete and ready for use. Delivery and support details are still being worked out. The system will be used "in house" for EXOSIM simulation work up until the time it is delivered.

2.5. New Developments

Simulation hardware in the DETL is under continuous development and improvement. All new developments center on parallel function processing concepts.

2.5.1. Developments Under Way

New developments currently under way are all incremental improvements to the existing hardware. For example, a Multibus II based PFP unit is currently being developed. The host interface, processing elements, and system busses will obviously be different, but the first MB II system will use the same crossbar, crossbar interfaces, and sequencer, as the exixting PFPs. After the first MB II system is complete, new design efforts on the sequencer and/or crossbar will intensify.

2.5.1.1. Multibus II Support

All PFPs built to date have used Intel's Multibus I as the main system bus. (The system bus provides the data path between all nodes and the host computer.) The bus, which has been in existence since the mid 1970's, has become very dated, and has lost some of its populartity. The newer, more powerful commercially available processor boards are now being built to newer, more powerful bus structures, such as the VME bus and Multibus II. Multibus II is a natural progression path for the PFP systems. Programs developed for Intel processor boards built to the Multibus I specification can migrate to the Multibus II boards with minimum modifications.

The first Multibus II based processors to be supported are Intel's iSBC386/120 (based on a 20 Mhz 80386/80387 processsor/coprocessor combination and Intel's iSBC486/125 (based on a 25 Mhz 80486 with math coprocessor integrated in the 80486). The Multibus II system is divided into Modular Processing Subsystems (MPS). Each MPS will be based on a 20 slot Multibus II card cage. Each cage must contain a Central Services Module, a Master CPU, a disk controller , and an interface to the host and other MPS units., (The first host/MPS interface will be ethernet, primarily because the networking software already exists. SCSI interface software is in development and has the potential of providing both the disk controller functions, and the Host/MPS interface.) Sixteen slots in each MPS are available for slave processors, I/O boards, graphics interfaces, mass memory, or any other board built to the Multibus II standard. Thus, the contents of each MPS is flexible and can be configured differently for specific applications. One MPS may contain sixteen slave processors, each with an individual interface to the crossbar. (The current crossbar interface causes the processor to occupy two card slots.) Another MPS may contain only one crossbar interface, and a mixture of analog and digital I/O boards and thus act as a kind of I/O subsystem.

The master CPU in each MPS is booted by the disk controller and runs the standard UNIX 5.0 operating system. The master CPU then boots each of the slave CPUs. Each slave is booted with Intel software called iRMK (Real time Multitasking Kernel). The kernel is not a full fledged operating system. It provides the basic functions needed for loading and running programs and also supports a debug mode. The code running on any slave processor can be debugged from the master, including single step, by invoking the right compiler options.

The driving force behind the Multibus II has been to pick up support for another set of commercially available state of the art boards. A second benefit is in the increased flexibility of the bus. In the Multibus I system, the bus was primarily limited to loading code and reading results. All communication between the host and any slave processor had to be initiated by the host. Direct communication between slave processors was not possible. Also, all mass storage was located at the host. In the MB II system, communication between any two boards within an MPS can take place over the bus (as well as the crossbar) at any point in a simulation. Disk storage is distributed on multiple disks in system, each of which can be accessed by any processor located within that MPS at any time. (The ultimate goal is to have any disk available to any processor within the system. This should be attainable when the SCSI work is completed.)

Currently, one MPS is functional. The unit contains four iSBC386/120 processors and one iSBC486/125 processor. Test programs exercising both the crossbar and bus interfaces have been compiled and successfully run. Plans for finishing the first MB II PFP are to have two MPS units each with sixteen slave processors, all communicating over one crossbar. Presently, the crossbar interface piggyback board interferes with the next card slot, so that only 8 target processors can fit in an MPS. The interface board

is being reworked so that it does not interfere with the next card slot, and 16 target processors can fit in one rack. The initial MPS/Host interface will be ethernet, which most likely be replaced with SCSI at a later date.

2.5.1.2 SCSI Interface Support

SCSI (Small Computer Systems Interface) is a common cost effective standard supported by many computer and peripheral device manufacturers. PFP support for SCSI has primarily been started for use as a host/MPS interface, but also has potential as the interface medium for a number of other applications. (The possible Lockheed LATS/Georgia Tech PFP interface tentatively scheduled at Arnold Engineering Development Center is one example.)

Preliminary development of the CPU to CPU software has been started. SCSI interface boards, each with two interface ports per board, have been installed. Communication has been eastablished with the board and SCSI commands loaded and executed. CPU to CPU communication has been established, thus verifying the approach. The software routines and drivers necessary to use the SCSI interface for loading and booting programs are being investigated.

2.5.2 Planned Developments

Planned developments will center on enhancing our ability to perform real time "hardware in the loop" simulations. The PFP will continue to be the center of development, but through the use of standard interfaces the unit will become more flexible, so that a specific configuration for a specific application can be arrived at by putting together components in a "leggo block" type fashion. In order to do this, our custom circuitry will continue to be complemented by the use of commercially available products and standards.

Our next upgrade, Multibus II support, is nearing completion. We will be able to use the same crossbar, sequencer, and crossbar interface with a state of the art bus.

2.5.2.1 New Crossbar

The next upgrade will be the crossbar/sequencer pair. The goals on the crossbar are increased speed and increased number of ports. The next crossbar will be able to handle a minimum of 64 processors (possibly 128) with each processor port able to send and receive data on the same cycle. Technologies being looked at for the next crossbar include commercially available fiber optics, commercially available gallium arsenide chips, and custom VLSI chips. The next crossbar will probably be using very high speed serial lines (possibly as high as 1 Gbit/sec) for data transfer, thus reducing the number of wires needed for each data path and making room for more interface ports.

2.5.2.2 New Sequencer

The goal on the next sequencer will be to increase the speed and number of processors it can handle to match (or exceed) that of the next crossbar and to add more flexibility to it to support a wider range of problems (particularly problems which are being investigated through PhD research here in the lab such as molecular dynamics, and compressible and incompressible Navier Stokes fluid flow equations). Functions that will definitely be supported will be 1) variable message length transfers and 2) branching ability within the sequencer. Complete dynamic communications (so that no definite communications must be known before the start of the simulation) will also be considered.

2.5.2.3 New Processor/Crossbar Interface

In order for commercial processors to work with a new crossbar/sequencer combination, a new crossbar interface will be needed. Something similar to the crossbar interface we have built to the iSBX port , which is a standard supported by many commercial vendors will be developed. Intel has come out with a new standard specification for piggyback boards called the MIX (Modular Interface eXtension) [6] that it will be using on its new processors. The MIX architecture supports 8, 16, and 32 bit data transfers, and supports a full 4 gigabyte address space. A MIX module can be stacked on a Multibus II processor motherboard and still only use one card slot for the pair. Our tentative plans are to use the MIX interface as the base for the next Crossbar/sequencer interface.

2.5.2.4 Futurebus+ Support

Both the Multibus Manufacturers Group (MMG) and the VME International Trade Association (VITA) are moving to support a new bus specification called the Futurebus+. The bus has recently been approved as an IEEE standard, IEEE 896.1 [7]. DEC, Sun, Intel, Motorola, Signetics, National Semiconductor, and a host of others are already pledging support for this bus [8] [9]. Also, the Navy's Next Generation computer Architecture program has already adopted the Futurebus+ as the basis for the future Navy Backplane Standard for all Navy mission critical computers [10]. The bus is still in its infancy, with components such as backplanes and bus interface silicon is just now beginning to show up on the market, but the projected popularity of the bus along with its high performance specifications cannot be ignored. Our present plans are to look at the bus in greater detail for possible future custom board designs, and to make sure we have a path for integrating Futurebus+ products into the PFP environment. Our planned upgrade path will be through Multibus II. A version of the specification is under preliminary development to have both busses in the same card cage, with Multibus II on one connector and Futurebus+ on the other [8]. This way the Futurebus+ products can be gradually intergrated with the Multibus II work presently under way, perhaps eventually replacing the Multibus II as the main system bus.

3. Schedule/Milestones

1. PFP Technical Data Package delivered.

2. Final version of PFP Operator's Manual complete.

3. Final version of PFP programmer's Manual complete.

4. KDEC PFP hardware complete.

5. KDEC PFP software complete.

6. First Multibus II PFP hardware complete.

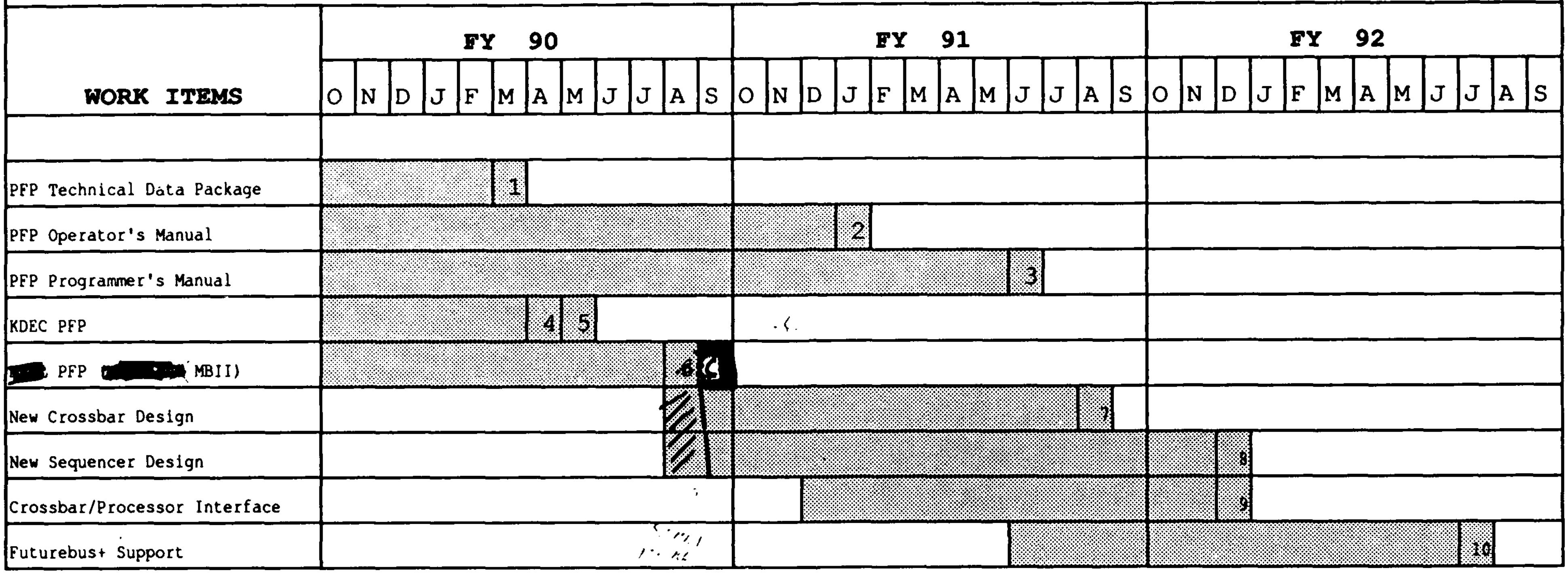
7. New crossbar design completed.

8. New sequencer design completed.

9. New crossbar/processor interface completed.

10. Futurebus+ based products supported.

FIGURE 23 [redacted] [redacted] SCHEDULE: PFP DEVELOPMENT



4. References

1. KEW Digital Emulation Laboratory Final Technical Report, Contract No. DASG60-85-C-0041, Volume 1, Georgia Institute of Technology, July 20, 1989.

2. Parallel Function Processor Technical Data Package, Contract No. DASG60-89-C-0142, Volumes 1-4, Georgia Institute of Technology, March 31, 1990.

3. Woods, Michael B., Parallel Function Processor Hardware Operation, Special Technical Report STR-0142-90-005, Contract No. DASG60-89-C-0142, Georgia Institute of Technology, January 23, 1990.

4. Pitts, Richard M., Parallel Function Processor Programmer's Manual, Special Technical Report STR-0142-90-006, Contract No. DASG60-89-C-0142, Georgia Institute of Technology, January 23, 1990.

5. Woods, Michael B, Richard M. Pitts, and James N. Hardage, Parallel Function Processor Reliability Test, Special Technical Report STR-0142-90-004, Contract No. DASG60-89-C-0142, Georgia Institute of Technology, January 23, 1990.

6. Intel Corporation, Multibus II I/O Product Lines, Intel Corporation., August 1989, pp 6 - 9.

7. Coco, Donna and Michele Clark, "Futurebus+ Logical Spec Shows at Buscon", EDN News Edition, Volume 35, Number 3A, Cahners Publications, February 8, 1990, pp 1, 20-21.

8. Curran, Lawrence, "Board Vendors Conquer Bus Limitations", Electronic Design, Volume 37, Number 3, VNU Business Publications, February 9, 1989, pp 31 - 40.

9. Andrews, Warren, "32 Bit Busses Contend for Designer's Attention", Computer Design, Volume 28, Number 21, Penwell Publications, November 1, 1989, pp. 78 - 96.

10. Naegele, Tobias, "Futurebus+ Comes on Strong", Military and Aerospace Electronics, Volume 1, Number 3, Sentry Publishers, March 1990, pp1, 34.